Cerebellum Based Car Auto-Pilot System

B. HSIEH, C.QUEK and A.WAHAB
Intelligent Systems Laboratory, School of Computer Engineering
Nanyang Technological University, Blk N4 #2A-32
Nanyang Avenue, Singapore 639798
SINGAPORE

Abstract: - Understanding the neural structures and the physiological mechanisms of the human brain and applying them onto practical applications is a difficult challenge. The practical realization of the sophisticated network of different brain components and the complex interaction between them to produce intelligent outcomes for practical applications is highly desirable. A model for a car auto-pilot system based on the physiological mechanism of one component of the human brain, the cerebellum, is proposed in this paper. An experimental validation of this model would be offered using a software-based simulator, which is further accredited with a hardware model that demonstrates the suitability and capability of adopting the cerebellum based approach for car maneuvering.

Key-Words: - Neural Networks, Cerebellum, Cerebellar Model Articulation Controller (CMAC), Auto-pilot

1 Introduction

Driving a vehicle is essentially based on the reaction of the driver to current road conditions that would affect any decision to accelerate, decelerate, stop or steer the vehicle. Speed of response is critical during vehicle maneuver as environmental conditions are continually changing and failure to react in time might lead to fatal consequences.

As such, a car autopilot system is basically a model of the response of the system to constantly changing environmental inputs. The physiological mechanism of the human brain in particular the motor control mechanism will be highly deemed as a good model for implementation of a car autopilot system. This paper will propose one such car autopilot system that utilizes the physiological mechanism of the human brain for motor control.

2 The Human Brain (Cerebellum)

The major components of the human brain basically consist of the cerebrum (the largest portion), the medulla oblongata, the thalamus, hypothalamus and the cerebellum.

Cerebellum works with muscles and the cerebral cortex to coordinate movement of skeletal muscles that will actuate voluntary and some involuntary movement of the body. An example is the vestibulocular reflex, i.e. when the head or body is moved, the cerebellum generates the motion commands for the eye muscles which make the eyes fixate on the same point. Early medical research [1] has given clear indications that the cerebellum is used for stable vertebrate control and that cerebellar lesions lead to instability of the motor system.

The cerebellum essentially functions by receiving sensory inputs from the cortex and generating outputs that coordinates and controls the motor systems. This gives rise to the suggestion that the cerebellum is in fact a very large associative memory [2] that simply associates a set of input conditions with a corresponding set of output controls.

3 Cerebellar Model Articulation Controller (CMAC)

The Cerebellar Model Articulation Controller [3] [4] proposed by Albus, based on the human memory models and neuromuscular control, is basically a neural network that models the structure and function of the cerebellum. The CMAC Neural Network employs table look-up methodologies whereby associations of input and output relationships are stored as the contents of memory cells of the CMAC.

3.1 Architecture of CMAC

The CMAC network is basically a form of associative memory that can be trained to implement non-linear functional mappings. Figure 1 shows a simplified 2-dimensional CMAC network which is scalable to n-dimensions.. For a 2D CMAC network, the CMAC memory can be visualized as a neural network consisting of a cluster of 2 dimensional self organizing feature map(SOFM) neural network.

However, instead of a random initialization of the neural net weights, they are fixed such that they formed a 2 dimensional grid CMAC as shown in the figure.

From Figure 1, the winning neuron in the CMAC memory at time k is identified as the neuron with weights $Q(y_1(kT))$ and $Q(y_2(kT - T))$ for inputs $y_1(kT)$ and $y_2(kT - T)$ respectively. The weights are effectively the coordinates i, j of the location of the neuron in the SOFM. The output of the winning neuron can be directly obtained from the weight $W_{i,j}$ of a particular winning neuron.

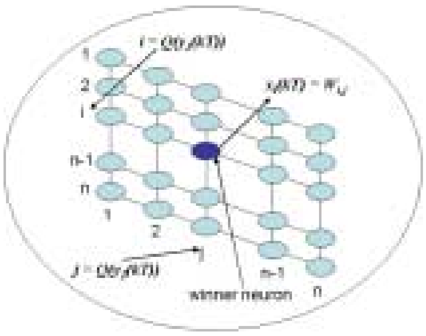


Figure 1: Architecture of a 2-Dimensional CMAC

An example of a trained 2-dimensional CMAC network used for braking control for reverse parking maneuvers is depicted in Figure 2 below.

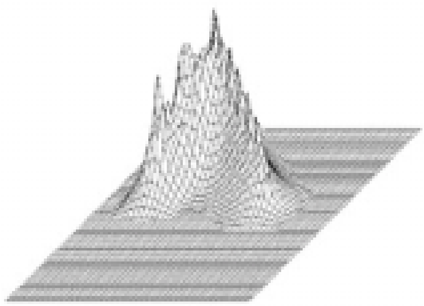


Figure 2: Typical 2-Dimensional CMAC Network Surface

4 CMAC for Vehicle Maneuver

Automated driving not only includes intelligent control of vehicle maneuvering in different road conditions like city driving, highway driving etc. but incorporates intelligent control for vehicle parking maneuvers as well. Parking maneuvering, unlike normal automated driving, is slightly more complex and complicated as parking maneuvers involve the frequent switching of forward and reverse gears, and maintaining a velocity ceiling of not more than 20kmh throughout the entire duration of the initiated parking sequence. In this paper, we propose a CMAC based controller that would perform intelligent vehicular parking maneuver. Currently, there exist two generic types of parking maneuvers, i.e. the Reverse Parking maneuver and the Parallel Parking

4.1 CMAC Controller (Reverse Parking)

Reverse Parking is one of the simpler and more straightforward parking methods. There exist 2 distinct manners whereby a reverse parking procedure could be executed. First of all, a simple one time reverse into the parking slot without any adjustment (Figure 3(a)) and the other requiring multiple-adjustments (Figure 3(b)) before fitting into the parking slot. The latter is performed when the road width is sufficiently wide to accommodate the entire parking procedure in a single reverse step.

(a) Without Parking Adjustment (b) With Parking Adjustment

Figure 3: Reverse Parking Techniques

The proposed reverse parking system algorithm using a CMAC controller could be illustrated in Figure 4 below.

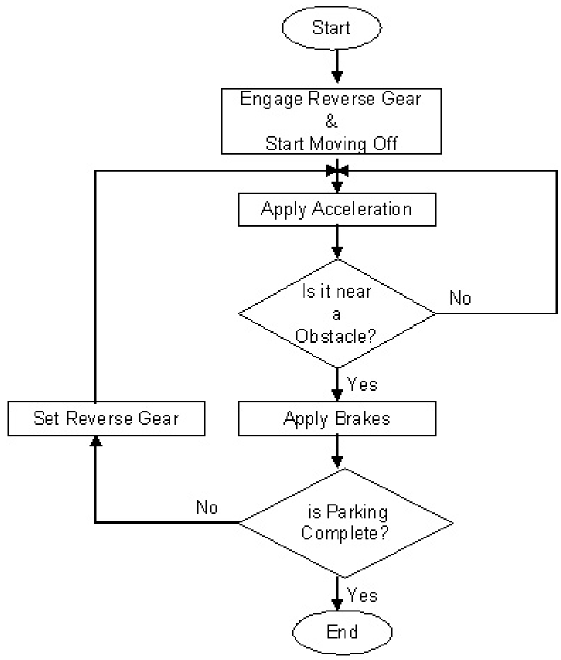


Figure 4: Reverse Parking Algorithm

Automated vehicle maneuver involves the coordination of steering control, gear control and velocity control (which is determined by the throttle and brake values). As such, the CMAC controller designed for this reverse parking system consists of 2 CMAC sub-controllers, one for steering control while the other to be responsible for velocity control. And the gear control will be determined by a decision "*black box*" which would initiate the switching of forward or reverse gear when certain conditions meted out are satisfied. Inputs to the CMAC networks are based on 8 strategically positioned distance sensors on the vehicle as shown in Figure 5.

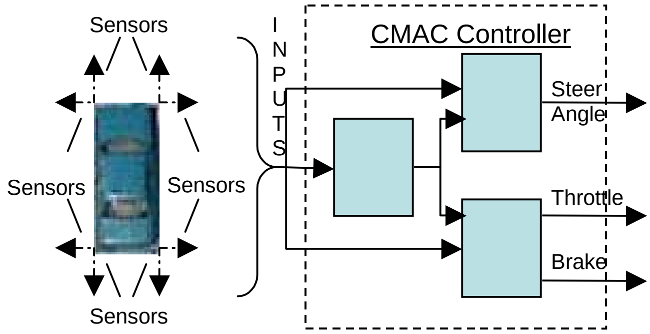


Figure 5: CMAC Controlled Vehicle

Steering Control in parking maneuvers is somewhat different from that of steering control in automated driving. The requirement to frequently engage and disengage the reverse gear requires 2 different steering angles to be generated for the same set of sensory inputs i.e. 2 different steering angles to be generated for the vehicle in a particular position on the track (with a particular set of sensory input data), one steering angle for the forward movement of the vehicle while the other for the reverse movement of the vehicle. Thus, besides the set of sensory input data, the CMAC Steering Sub-Controller requires the gear input to distinguish between the steering angle computed for forward driving and the other for reverse driving.

The CMAC Sub-Controller for steering of a vehicle in reverse parking is illustrated in Figure 6.

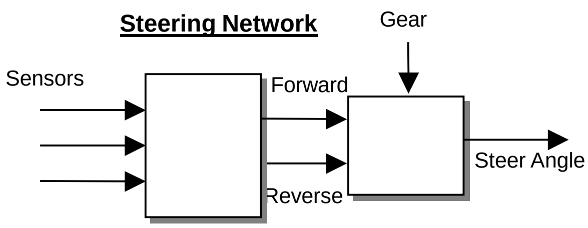


Figure 6: CMAC Block Diagram for Steering Sub-Controller

During the entire parking procedure the speed of the car is to be limited to a maximum value of 20kmh as a parking procedure involves refined slow maneuvers to enable the car to fit into the parking slot.

Similar to the case of automated driving, the TPS and Brake controls are antagonistic to the other. The vehicle accelerates its speed up to the maximum allowed value of 20kmh and stops when it approaches obstacles or barriers before reversing its direction of movement. Thus, the inputs to the TPS and Brake CMAC sub-controllers include both speed and sensory values.

In order to detect the presence of obstacles or barriers, the minimum distance to obstacles or barriers in the direction of motion must be computed. For forward motion, the sensory data the car must sense for obstruction are the front sensors while for reverse motion the set of sensory data required by the car include the back sensors. The 2 sets of sensory data as described above are mapped into a single valued input for the CMAC controllers by virtue of simply computing the minimum values from each set of data. Another input the CMAC controllers required is the current velocity of the vehicle which is a feedback input.

The CMAC Sub-Controller for throttle(TPS) and braking control of a vehicle in reverse parking is illustrated in Figure 7.

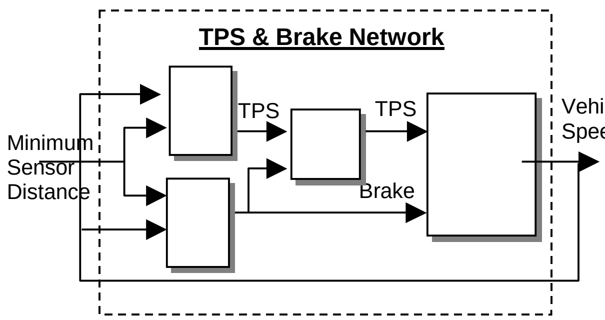


Figure 7: CMAC Block Diagram for TPS/Brake Sub-Controller

The CMAC Controller proposed above was implemented using a software car simulator [5] which is used to simulate a car maneuvering sequence in a virtual driving environment.

4.2 Performance of CMAC Reverse Parking

The performance of the intelligent control for the implemented Reverse Parking System could be measured by the qualitative measure that depends on the final position of the vehicle in the parking lot. The

Quality of Parking (QoP) measure is measured by the 3 sensors indicated in Figure 8. The 3 sensors involved in the computation of the QoP measure are essentially the [1]back left to barrier sensor, [2]back right to barrier sensor and [3] the left back to side sensor (for left parking) or the right back to side sensor (for right parking).

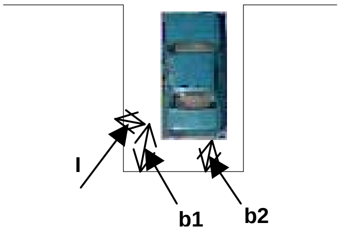


Figure 8: Reverse Parking Quality Components

The computation for QoP measure requires the computation of the error in the final parking position of the vehicle as compared to a desired position. The error ,ε, is calculated as follows:

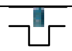
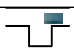
$$\varepsilon = \sqrt{\frac{(l_{final} - l_{desired})^2 + (b1_{final} - b1_{desired})^2 + (b2_{final} - b2_{desired})^2}{3}} \qquad \text{Eq (1)}$$

The max error achievable in the above equation is a value ‘1’ while the min. error is of value ‘0’.
The QoP measure is determined by the following equation:

$$QoP = \varepsilon_{max} - \varepsilon_{current} \qquad \text{Eq (2)}$$

The CMAC Network (with the CMAC sub-controllers) for reverse parking were trained with a collated data set that comprises of different sets of human-driven parking sequences that originate from the initial starting positions of the 6 test cases as indicated in Table 1 below which tabulates the Reverse Parking System’s performance results according to the quality of parking (QoP) specified by Eq(2)

Table 1: Reverse Parking Test Results

Test Case	Error. ε	QoP
1st 	0.221	0.779
2nd 	0.189	0.811
3rd 	0.103	0.897
4th 	0.182	0.818
5th 	0.233	0.767
6th 	0.192	0.808
Average	**0.186667**	**0.813333**

As shown in Table 1 above, the CMAC controlled reverse parking system is capable of achieving high quality parking maneuvers with close to perfect parking performance (>80% perfect parking). A video documenting a successful parking sequence can be found here.

4.3 Hardware Implementation

With the successful implementation of the software simulation for autonomous driving using CMAC technologies, the realization of the software simulator onto an actual hardware implementation follows. A remote controlled (RC) (shown in Figure 9) model car will be used in this hardware implementation.

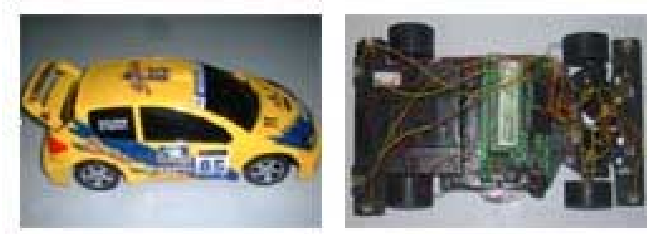


Figure 9: RC Model Car

The hardware RC Model Car used would not be hosting the CMAC network; instead an on-board microcontroller would be serving as a transmission link between the RC model car and a Host PC. The Host PC would be carrying the CMAC network and would carry out all the necessary computational functions that are required in an intelligent automated control of the vehicle. The Host PC receives all the sensory data from the model car via its on-board microcontroller, computes the necessary control signals for the car and outputs these control signals back to the model car via the same transmission link that actuates the DC and servo motors resulting in the desired motion of the vehicle.

Figure 10 below shows the architecture of the interface between the software simulator and the hardware model car

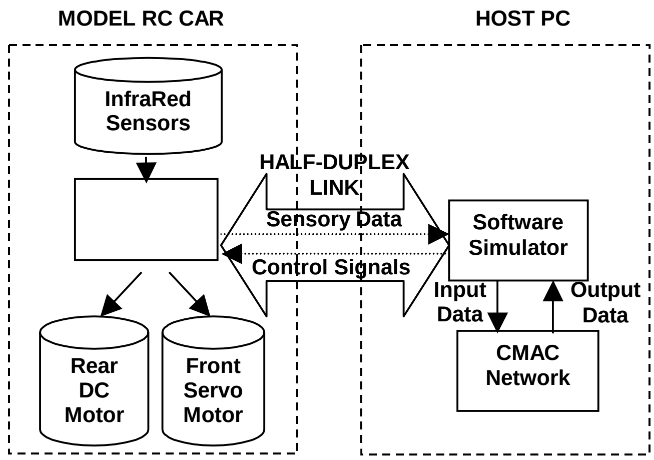


Figure 10: Software-Hardware Interface

With reference to the Figure 9**,** when intelligent maneuver of the model car is initiated, the on-board microcontroller would immediately retrieve sensory data from all its 8 Infrared sensors on-board. The microcontroller would then packed the 8 different values obtained into a vector comprising of all 8 values and transmit via the half-duplex transmission link to the Host PC. The Host PC upon receipt of this sensory information would process the received information and present the appropriate information as input data to the CMAC network. The CMAC Network processes the input data and outputs a set of control signals namely the 2 basic control signals for controlling the front servo motor and the rear DC motor. The Host PC would then transmit the control signals back to the microcontroller via the same half duplex link. The microcontroller would then actuate the front servo and rear DC motors based on the received control signals.

Video documentary of the success of the hardware implementation could be found here.

4 Conclusion

A new approach to intelligent car maneuvering has been proposed and proven with the implementation of the cerebellum based car auto-pilot system.

The current adopted approach based on the associative relationship between a set of input conditions and the corresponding set of output controls have successfully demonstrated the ability to effectively control and maneuver a vehicle without any human intervention and at the same time achieving a high standard of success.

The CMAC network utilized in the implementation is simply an associative memory network that performs low-level cognitive task of simply associating inputs with known outputs. Proposed future work will be on development of systems or networks capable of high-level cognitive tasks such as reasoning, symbol processing etc that would inevitable translate into better performance and higher intelligence.

References:

[1] G. Holmes, The cerebellum of man, *Brain*, 1939, pp. 1-30.

[2] P. Kanerva, Associative-Memory models of the Cerebellum, *Artificial Neural Networks*, 1992, pp 23-34

[3] J.S. Albus, A new Approach to Manipulator Control: The Cerebellar Model Articulation Controller(CMAC), *Transactions of the ASME Journal of Dynamic System,Measurement and Control*, Vol 97, 1975, pp 220-227

[4] J.S. Albus, A Theory of Intelligent Systems, *Proceedings 5th IEEE International Symposium*, Vol 2, pp 866-875

[5] P. Michel, H.C. Quek, and M. Toh, Fuzzylot: A Self-Organising Fuzzy Neural Rule-based Pilot System for Automated Vehicles, *Neural Networks Journal*, Vol 14, No 8, 2001, pp1099-1112